

1

CLAIMS

What is claimed is:

- 1 1. A computer-implemented method for optimizing an executable program having a
2 plurality of functions and at least one function with a first name associated with executable
3 code that implements the function at a first address and at least one linkage stub code
4 segment having code that branches to the first address and a symbolic name by which the
5 function is invoked in the program, comprising:

6 identifying branch instructions having target addresses that reference the linkage
7 stub code segment; and

8 replacing the target addresses of the branch instructions with the first address.

- 1 2. The method of claim 1, further comprising replacing the target address of the
2 branch instructions with the first address only in functions that are reached during program
3 execution.

- 1 3. The method of claim 1, further comprising:
2 searching a symbol table for an entry having a symbolic name that that is a
3 derivation of the first name and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

- 1 4. The method of claim 1, further comprising:

10012768-1

2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore prefix and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 5. The method of claim 1, further comprising:

2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore suffix and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 6. The method of claim 1, further comprising:

2 replacing function entry points in the executable program with breakpoints,
3 whereby breakpointed functions are generated; and
4 upon encountering a breakpoint of a breakpointed function during program
5 execution, identifying within the breakpointed function branch instructions that target
6 linkage stub functions.

1 7. The method of claim 6, further comprising:

2 storing original instructions from the function entry points prior to replacement
3 with the breakpoints;

4 upon encountering a breakpoint of a breakpointed function during program
5 execution, restoring the original instruction to the entry point of the breakpointed function.

1 8. The method of claim 6, further comprising:
2 searching a symbol table for an entry having a symbolic name that that is a
3 derivation of the first name and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 9. The method of claim 6, further comprising:
2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore prefix and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 10. The method of claim 6, further comprising:
2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore suffix and reading a linkage stub address associated with the
4 symbolic name; and

5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 11. The method of claim 1, further comprising:
2 replacing entry points of linkage stub code segments in the executable program
3 with breakpoints, whereby breakpointed linkage stubs are generated; and
4 upon encountering a breakpoint of a breakpointed linkage stub during program
5 execution, changing a target address of a branch instruction that branched to the
6 breakpointed linkage stub to reference the function referenced by the breakpointed linkage
7 stub.

1 12. The method of claim 11, further comprising:
2 searching a symbol table for an entry having a symbolic name that that is a
3 derivation of the first name and reading a linkage stub address associated with the
4 symbolic name; and
5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 13. The method of claim 11, further comprising:
2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore prefix and reading a linkage stub address associated with the
4 symbolic name; and

5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 14. The method of claim 11, further comprising:

2 searching a symbol table for an entry having a symbolic name that matches the
3 first name with an underscore suffix and reading a linkage stub address associated with the
4 symbolic name; and

5 replacing target addresses of branch instructions having target addresses equal to
6 the linkage stub address with an address at which the code that implements the function is
7 stored.

1 15. An apparatus for optimizing an executable program having a plurality of functions
2 and at least one function with a first name associated with executable code that
3 implements the function at a first address and at least one linkage stub code segment
4 having code that branches to the first address and a symbolic name by which the function
5 is invoked in the program, comprising:

6 means for identifying branch instructions having target addresses that reference the
7 linkage stub code segment; and

8 means for replacing the target addresses of the branch instructions with the first
9 address.